Complexity Issues in Data Intensive High End Computing

Alok Choudhary, Professor
Director: Center for Ultra-Scale Computing and Information Security
Dept. of Electrical & Computer Engineering
And Kellogg School of Management
Northwestern University
choudhar@ece.northwestern.edu

Data Intensive Computing?

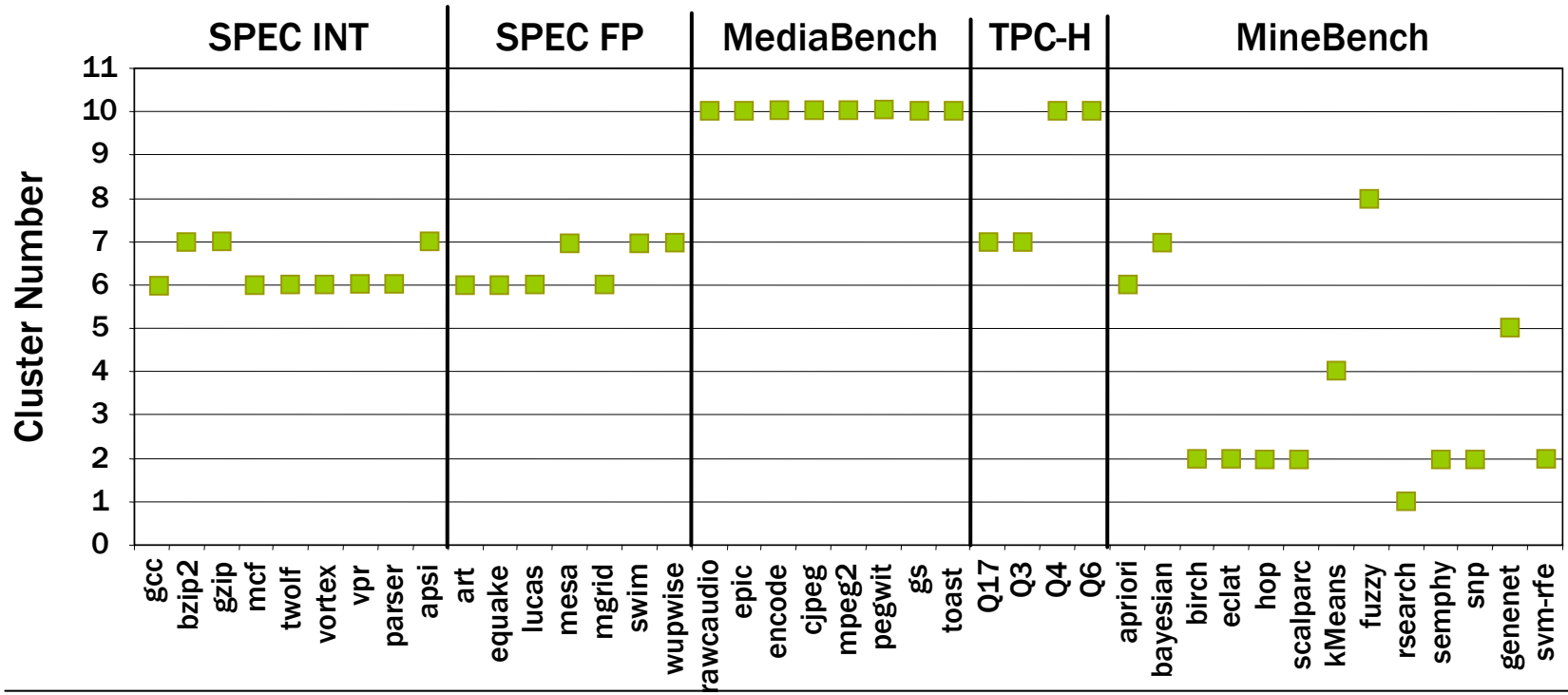
- Derive “actionable knowledge” or “insights” from massive data?
- Data drives computing?
- More computing (than “usual applications”) per data item?

NU-MineBench

Algorithms	*Category*	*Description*
k-Means	Clustering	Mean based data partitioning method
Fuzzy k-Means	Clustering	Fuzzy-logic based data partitioning method
BIRCH	Clustering	Hierarchical data segmentation method
HOP	Clustering	Density based grouping method
Naive Bayesian	Classification	Statistical classifier
ScalParC	Classification	Decision tree based classifier
Apriori	ARM	Horizontal database, level-wise mining based on Apriori property
Eclat	ARM	Vertical database, equivalence class based method
SNP	Bayesian Network	Hill-climbing search method for DNA dependency extraction
GeneNet	Bayesian Network	Microarray based structure learning method for gene relationship extraction
SEMPHY	Expectation Maximization	Phylogenetic tree based structure learning method for gene sequencing
Rsearch	Pattern Recognition	Stochastic Context-Free Grammar based RNA sequence search method
SVM-RFE	Support Vector Machines	Recursive feature elimination based gene expression classifier
PLSA	Dynamic Programming	Smith Waterman optimization method for DNA sequence alignment

Data Intensive - Data Mining

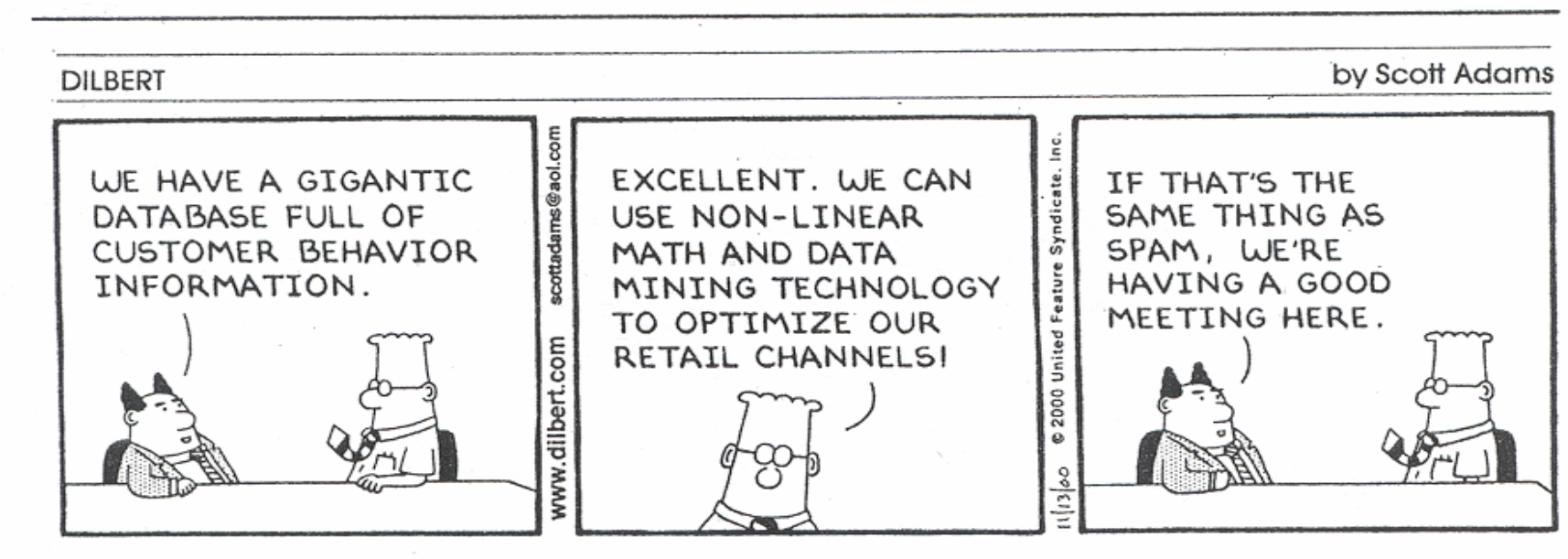
- 25 dimensional performance and characterization data. Mining used to cluster
- NU MINEBENCH
- http://cucis.ece.northwestern.edu/projects/DMS/MineBench.html



Example Differences in Observed Metrics

****Parameters shown are "Per Instruction" values**

Parameter	Benchmark of Applications				
	SPECINT	SPECFP	MediaBench	TPC-H	Data Mining
Data References	0.807	0.550	0.568	0.483	1.565
Bus Accesses	0.007	0.012	0.001	0.005	0.043

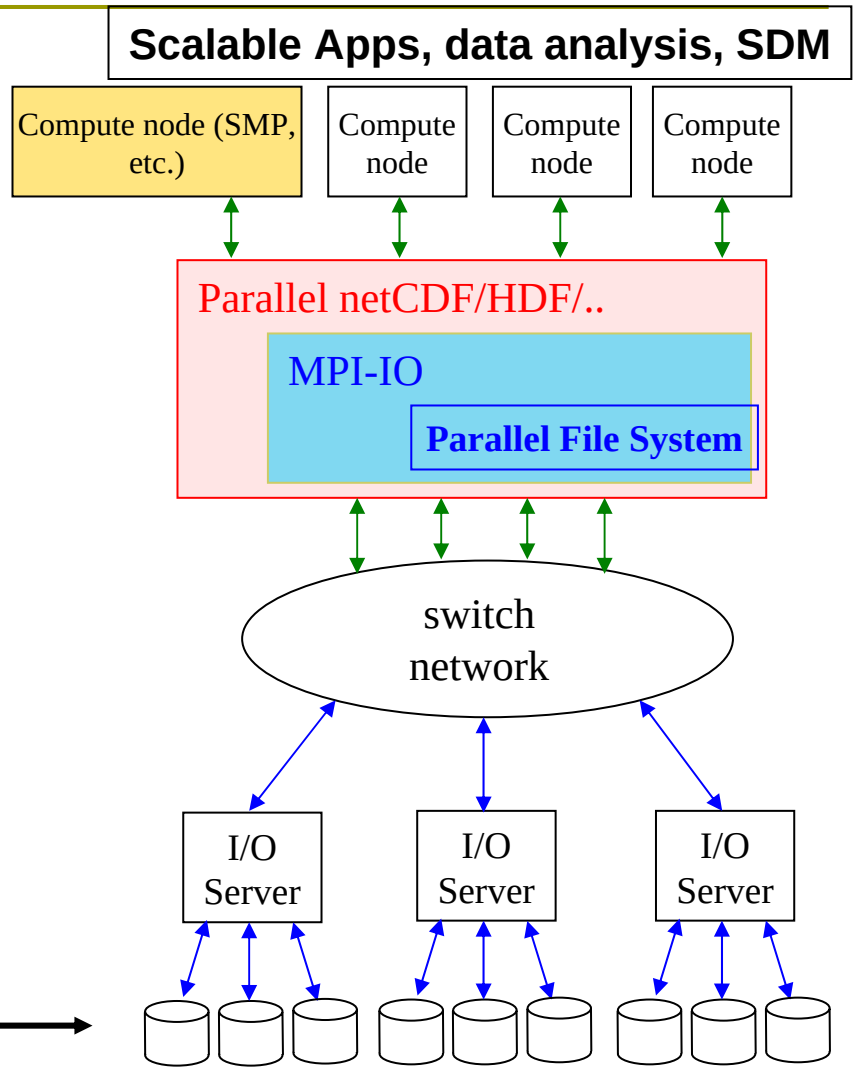


Complexity Issues

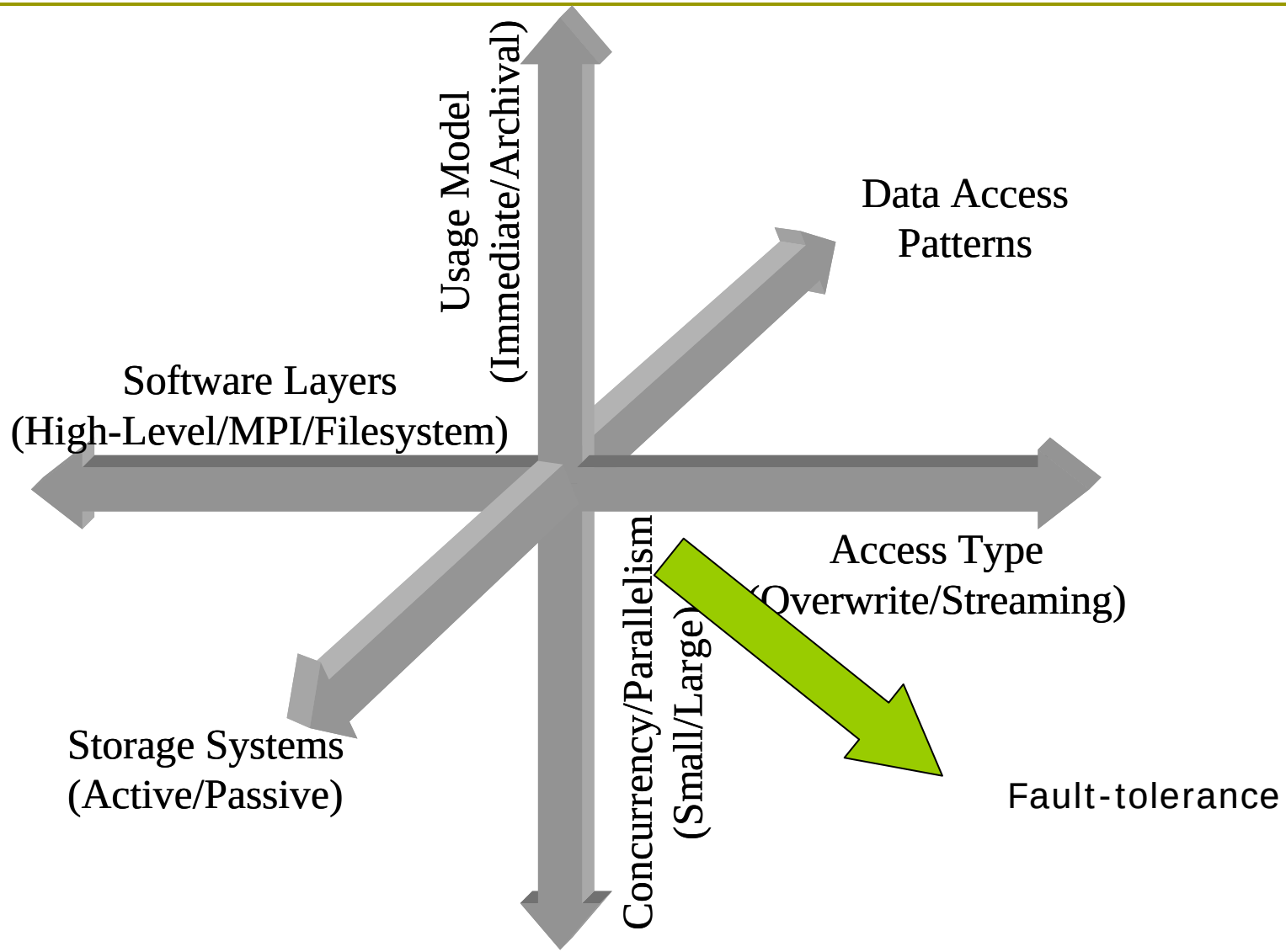
User Perspective: It is already very complex!

User

- Based on a lot of current apps
- High-Level
  - E.g., NetCDF, HDF
  - Applications use these
- Mid-level
  - E.g., MPI-IO
  - Performance experience
- Low Level
  - E.g., File Systems
  - Critical for performance in above



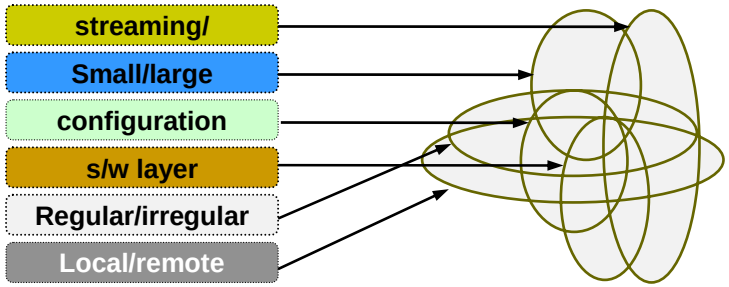
Some Complexity Dimensions



HEC File System

User Burden

Complex non-portable optimization space

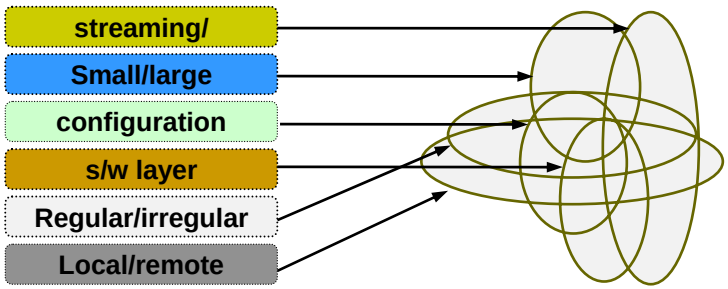


- **Ineffective interfaces**
- **Non-communicating layers**
- **Non-portable**

- Next decade
- BW 10X? Yes in spec, no in observed
- Spindle count – don't care
- Concurrency 10X? Yes, but don't want to see it
- Seek Efficiency – too low level for user to even think about?
- Failures – User should not see them!

Make it a Service

Current



- user burdened
- Ineffective interfaces
- Non-communicating layers

Goal